Author(s): Vihlman, Mikko & Hyyti, Heikki & Kalmari, Jouko & Visala, Arto

Title: Detection and species classification of young trees using machine perception for a semi-autonomous forest machine

Year: 2015

Version: Post print

Please cite the original version:

Vihlman, Mikko & Hyyti, Heikki & Kalmari, Jouko & Visala, Arto. 2015. Detection and species classification of young trees using machine perception for a semi-autonomous forest machine. IEEE International Conference on Robotics and Automation (ICRA). 6 p. DOI: 10.1109/icra.2015.7139394.

Detection and Species Classification of Young Trees Using Machine Perception for a Semi-Autonomous Forest Machine*

Mikko Vihlman[1], Heikki Hyyti[1], Jouko Kalmari[1] and Arto Visala[2]

***Abstract*— An approach to automatically detect and classify young spruce and birch trees in forest environment is presented. The method could be used in autonomous or semi-autonomous forest machines during tending operations. Detection is done by segmenting laser range images formed by a rotating laser scanner. Classification is done with a two-class Naive Bayes classifier based on image texture features. Multiple combinations of 99 features were tested and the best classifier included eight features from the co-occurrence matrix, local binary patterns, statistical geometrical features and Gabor filter. 79% of spruces and birches in the testing material were detected and 74% of these were correctly classified. Results suggest that the approach is suitable but there are still some challenges in each of the processing steps. Iteration between segmentation and classification is needed to increase reliability.**

I. INTRODUCTION

Selective cleaning is an important silvicultural tending operation. It improves growing conditions of certain trees by removing other trees and the surrounding vegetation. The starting point in cleaning is to analyze the forest, i.e. detect trees and recognize at least the target species. Robotics and automation is one approach for enhancing the operation and reducing the related costs. Fig. 1, for instance, shows the semi-autonomous forest machine used by Hyyti et al. [1] for real-time detection of spruce seedlings during automated mechanical point cleaning. The same machine with a different processing head is used for tending of slightly older trees. In the present work the aim is to detect trees of 3–5 meters tall and classify them as spruce or birch. No movement of the forest machine is considered; all data is processed off-line.

Some research has already been done on detecting and classifying trees in camera images. Haering and da Vitoria Lobo [2], for instance, detected deciduous trees with texture analysis. Results were reasonable but they focused on separating vegetation from other objects such as buildings and other landforms. This may not be enough for selective cleaning since the environment includes mainly vegetation and it is important to detect each individual tree. Sampsa Kosonen [3] used color and texture analysis to classify tree trunks of pine, spruce, birch and aspen. Similarly, Ali et al. [4] detected tree trunks during autonomous navigation. These studies reported success rates of over 80%. Young trees, however, do not have as visible trunks as mature trees, making the approaches inapplicable as such.

Fig. 1. Forest machine used as a research platform for automated mechanical point cleaning of tree seedlings.

Many works have utilized aerial 3D laser scanning and multispectral imaging to detect trees, e.g. [5] and [6]. The problem with this approach, however, is to concentrate on landscape information. Tree maps obtained are not accurate enough for forest machines which also need a way to frequently update the map. Spectral imaging still appears interesting considering species classification. Jääskeläinen et al. [7] successfully distinguished between coniferous and deciduous trees using spectrometer measurements. They observed that reflectance spectra start to differ close to infra-red wavelengths. This raises the idea of using cameras instead of spectrometers since many regular machine vision cameras can detect wavelengths of light up to about 1000 nanometers. Jääskeläinen et al. [7], however, mention that individual spectra vary notably. Geographical location, annual rhythm, tree age and pollution have also been noted to affect spectra [8].

Laser distance measurements provide a natural way of obtaining accurate information about actual tree locations. Visa Jokelainen [9] used a 3D laser scanner and a camera to detect and localize young trees. The scanner detected candidates for treetops and false positives were removed by finding vertical features, i.e. tree trunks, in camera images. Similarly, Erikson and Vestlund [10] detected tree trunks by finding

*This work was done under project NeoSilvix funded by TEKES, the Finnish Funding Agency for Technology and Innovation, and participating companies.

[1]M. Vihlman, H. Hyyti and J. Kalmari are PhD students with the Autonomous Systems research group, Department of Electrical Engineering and Automation, Aalto University School of Electrical Engineering, PO Box 15500, FIN-00076 Aalto, Finland {mikko.vihlman, heikki.hyyti, jouko.kalmari}@aalto.fi

[2]A. Visala is a Professor with the Department of Electrical Engineering and Automation, Aalto University School of Electrical Engineering, PO Box 15500, FIN-00076 Aalto, Finland arto.visala@aalto.fi

vertical features in laser range images and laser intensity images. These studies share the problem that vertical features of young trees may not be visible. Still, they demonstrate benefits of combining depth information of the laser scanner with more traditional image-based detection. As is seen in Fig. 4, range images provide a clear representation of the forest, at least when trees are distinct. There has been some research on segmentation of laser range images, e.g. [11] and [12]. The methods often work well for structures that are easily modeled, which is not the case in tree detection.

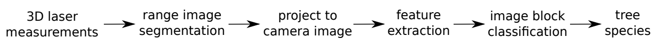

Fig. 2. Processing steps in tree detection and classification.

This work uses the approach presented in Fig. 2. Trees are detected with a 3D laser scanner and classified using camera images. Detection is done by segmenting laser range images. Tree regions are then projected to the camera image, divided to image blocks and classified using a two-class Naive Bayes classifier based on image texture features. The aim is to keep methods as simple as possible and to show that the concept works. As shown in Fig. 2, the order is strictly from segmentation to feature extraction and classification. The laser scanner is not used for classification and camera images are not used to refine segmentation results.

The work is structured as follows. Section II begins by describing the measurement system, its calibration and the data used for testing the system. It then presents the main methods needed for segmentation and classification. Results are presented in Section III and discussed further in Section IV.

II. METHODS

A. Measurement System and Data

Fig. 3 shows the measurement system. The system includes the self-designed 3D laser scanner used in [9]. It is a 2D scanner (SICK AG LMS111) continuously tilted up and down to construct the 3D point cloud. The system has a color machine vision camera (NET GmbH Foculus FO422C) fixed to a metal plate connected to the base of the scanner. The camera has a Schneider Kreuznach Cinegon 1.8/4.8 lens giving a wide angle of imaging. There is also a second camera in the system, with the purpose of taking infra-red images. However, it is not used in this work.

While collecting data, the measurement system is pointing slightly downwards fixed on a pole of about three meters tall (the system is resting close to ground level in Fig. 3). The system and the measurement computer are attached on an all-terrain vehicle which allows moving the whole system easily around the forest. Actual processing of the data is done off-line on a separate computer.

Calibration allows projecting laser range measurements to camera images. The camera is first calibrated separately using the Matlab implementation by Bouguet [13]. The camera model describes the mapping of three-dimensional points to two-dimensional pixels in the camera image and

Fig. 3. The measurement system with a rotating laser scanner and two machine vision cameras.

is similar to the model by Heikkilä [14]. The rotation and translation between the camera and the scanner is then found by imaging a calibration grid in multiple positions with both devices at the same time. Initial parameters are found with the stage I of the process by Unnikrishnan and Hebert [15]. Final parameters are obtained by iteratively minimizing the distance between the center point of the calibration grid in the camera image and the corresponding point projected to the image from the scanner coordinate system.

Data was collected in Janakkala, Finland, in the beginning of September 2011. Trees were mostly spruce and birch, about 3–5 meters tall. Some of the deciduous trees showed early autumn foliage. Data consist of laser scans and images of 13 locations in a small region of a forest. One of the locations was neglected due to some overlapping between the images. The other 12 locations were split randomly in half to have separate data for training and testing.

B. Tree Detection and Localization

Tree detection and localization are done by segmenting laser range images. The aim is to represent each tree as a cloud of 3D points before species classification. Segments could also be used to map tree locations and to compute tree-specific properties such as the volume, height and diameter. Segmentation applies the following algorithm (see Fig. 4) to search points placed uniformly around the range image:

0) Take a search point $\vec{p} = [p_x, p_y]^{\mathrm{T}}$ in the range image. Assume that it belongs to a tree.
1) Collect the depth value d_p at $\vec{p}$. Make a binary image of the same size as the range image. The pixel at location $\vec{p}_2$ is $\{1\}$ if the corresponding depth value d_2 satisfies $|d_p - d_2| \leq d_{\mathrm{th}}$ (with threshold d_{th}), and $\{0\}$ otherwise.
2) Find all points with value $\{1\}$ in the binary image connected to the search point $\vec{p}$.
3) If the number of connected points is at least n_{th}, collect the segment and remove it from the range image. Repeat the algorithm with the next $\vec{p}$ until all points have been processed or subsumed by other segments.

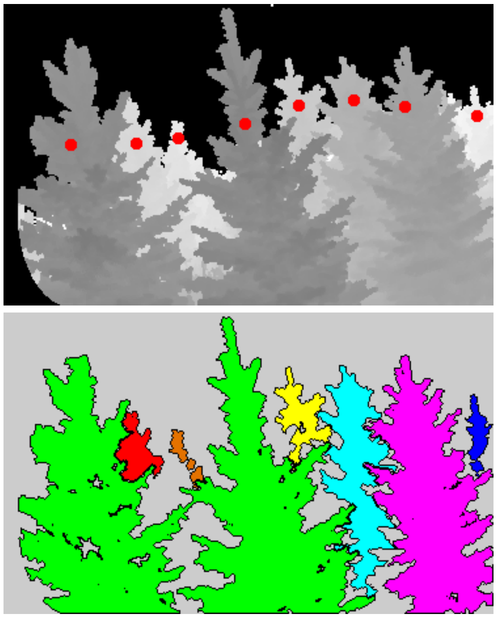

Fig. 4. Segmentation of trees. Original range image (top, shown only partly) with eight search points (red circles), and the segmentation result (bottom) with color coding. Search points are here manually marked and expanded for illustration. The fourth search point from the left is effectively removed by the first segment (the two trees have connected branches). The second tree from the left is occluded by a branch of the tree on the left and thus not fully connected to the segment of the search point.

The algorithm should succeed in finding most trees with suitable parameters d_{th} and n_{th}. This work sets d_{th} to 500 millimeters and n_{th} to 500 points. A lot of over-segmentation and under-segmentation is expected due to the simplicity of the algorithm. There is no attempt to solve over-segmentation. Each tree is allowed to be represented by multiple segments since this should not notably affect classification. Under-segmentation may be a problem, however, since features get mixed during classification when segments include multiple trees or the surrounding ground.

Neglecting ground and smaller vegetation around the trees is done by detecting the direction the tree grows in. Using the coordinates of the segment, principal component analysis (PCA) finds three coordinate axes in the order of diminishing variance. The axis corresponding to the height axis of the scanner estimates the direction of growth, others form the horizontal plane. To reduce the effect of ground points, PCA axes are formed iteratively by removing the lowest $p\%$ of the segment at each iteration. Finally, the distribution of the highest points in the horizontal PCA plane defines a threshold window for removing surrounding points from the segment.

PCA axes are also useful for detecting segments containing only ground points (instead of a tree surrounded by ground). Segment is considered ground-only 1) if it spans a longer distance along both horizontal axes compared to the distance along the height axis, or 2) if it spans at least 60% higher distance along one horizontal axis compared to the distance along the height axis. These rules were formed experimentally with the training material and emphasize that ground segments are relatively wider than tree segments.

C. Feature Extraction

After segmentation, tree segments are transformed from the scanner coordinate system to image pixels and divided to rectangular image blocks. The width and height of the block vary slightly to cover the segment evenly but are about 40 pixels. Texture features from the following eight groups are computed for each image block:

- 20 descriptors of the Gray Level Co-Occurrence Matrix (GLCM) [16]: contrast, correlation, energy, homogeneity and entropy in four directions.
- 44 descriptors of the Gray Level Run Length Matrix (GLRL) [17]: 11 features in four directions.
- 2 descriptors of Edge Frequency: the number of edge pixels per unit area using Roberts method and the zero-cross Laplacian of Gaussian.
- 4 Fractal Dimension descriptors [18].
- 16 Statistical Geometrical Features [19].
- 2 descriptors of Local Binary Patterns (LBP) [20]: mean and standard deviation of the rotation invariant LBP histogram.
- 7 descriptors of the three-level Wavelet Decomposition using Daubechies db1 wavelets: three measures of energy in the wavelet approximation and detail images, and the four rotation invariant features described in [21].
- 4 rotation invariant Gabor Filter descriptors [21].

For GLCM and GLRL features, image is first scaled to have eight numerical levels. Not all features listed above are used in the final classification; instead, the classifier is trained with multiple combinations and the one with the best result is chosen.

D. Classification

Classifying the tree species is done in two steps. First, a two-class Naive Bayes classifier is utilized to compute the probability density function of class label L (spruce or birch) given the feature vector $\vec{F}$ of the image block. The naivety comes from assuming that features are independent of each other. This simplifies the Bayesian posterior model to

$$p(L|\vec{F}) = \frac{p(L)}{p(\vec{F})} \prod_{i=1}^{n} p(F_i|L), \tag{1}$$

where n is the number of features and F_i is the feature number i. Function $p(L)$ describes the prior knowledge about the prevalence of each species. Reflecting ignorance, classes are assumed equally probable, i.e. $p(L) \propto 1$. Function $p(\vec{F})$ is a normalizing term. Function $p(F_i|L)$ is the likelihood of feature F_i given that the image block represents class L. It is estimated directly with feature histograms formed from the training data in order to avoid unnecessary assumptions about individual feature distributions.

Bayesian approach is used due to its simplicity and expandability. Additional tree species can be considered simply by adding new class labels to functions $p(L)$ and $p(F_i|L)$. New explanatory features can be added by including one more function $p(F_i|L)$ in the product for each new feature. Assuming features independent also makes it straightforward to build a more complex distribution for some feature or a joint distribution for some group of features while utilizing simple models for the other features. Function $p(L)$ can take into account that each forest is different and that the forest machine changes the environment while cutting down trees.

In the second classification step, each image block is given value ± 1 depending on which of the two classes has a higher density. The whole segment is then classified based on the density-weighted sum of block classes.

E. Training

As mentioned in Section II-A, there were six images for training and six images for testing. Cross-validation was used for choosing and training the best classifier. Trees in the six training images (28 spruces, 24 birches) were manually marked and labeled. Each training image was classified by neglecting it while fitting the classifier with the other five training images. This was repeated with multiple feature combinations to find the combination giving the best result.

The best classifier included eight features from the following feature groups: co-occurrence matrix, local binary patterns, statistical geometrical features and Gabor filter. It correctly classified 93% of spruces and 88% of birches in training images. The classifier was finally trained with all six training images before processing the actual testing images.

III. RESULTS

The previous section presented the main steps in training, segmentation and classification. To summarize, tree segments are first searched in the laser range image. Each segment is then projected to the camera image and divided to image blocks. Image blocks are classified using a two-class Naive Bayes classifier and the final classification is done based on probability densities of the block classifications. Figs. 5–8 show these steps for one of the six testing locations.

The segmentation algorithm managed to find most trees, but there were many split and combined segments and some trees were found only partly. As Fig. 5 shows, ground segments were filtered out quite effectively, but some tree segments still contained ground after segmentation. Due to imperfect segmentation, the following rules were used to estimate success rates: 1) Segments belonging mostly to a single tree were considered a single segment if at least about a half of the actual tree was found. 2) Segment containing two trees, one big and one small, was considered a single segment belonging to the species of the larger tree. The smaller tree was not considered as detected or undetected, it was interpreted as not being present. 3) Segments with multiple trees of similar size but of different species were considered incorrect segmentations and market as undetected.

Table I shows detection and classification rates for all testing images. Classification results are relative to the number of trees detected. Rates include only trees visible in camera images, that is, about 40% of segments found by the laser scanner. Detection and classification rates were similar both for spruce and birch. The overall correct detection rate was 79% and the overall correct classification rate was 74%. Total success rate was thus 58%.

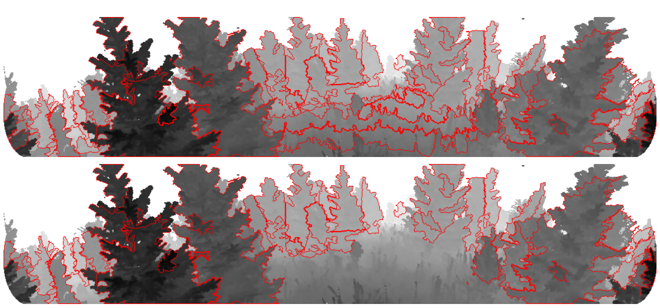

Fig. 5. Segment boundaries (red curves) for image number 4 before (top) and after (bottom) removing ground segments. See Figs. 6, 7 and 8 for the classification of the trees in this location.

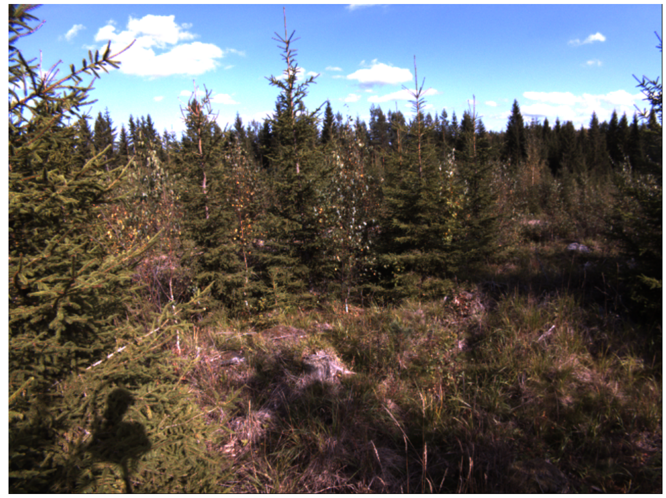

Fig. 6. Image number 4. See Fig. 5 for the corresponding laser range image, Fig. 7 for the image with block classifications and Fig. 8 for the image with final classifications.

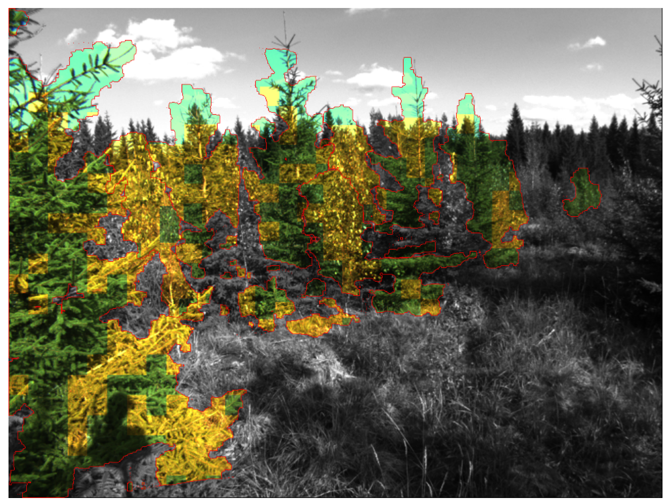

Fig. 7. Block classification for image number 4 (see Fig. 5 for the corresponding laser range image and Fig. 6 for the original camera image). Red curves show the segment boundaries. Block classifications describe which parts of the tree segments resemble spruce (green color) and which resemble birch (yellow color). See Fig. 8 for the final classification based on probability densities of the block classifications.

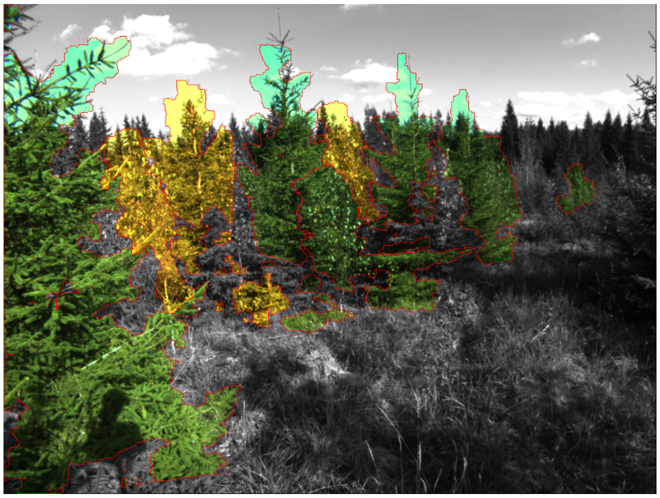

Fig. 8. Final classification for image number 4 (see Fig. 5 for the corresponding laser range image and Fig. 6 for the original camera image). Red curves show the segment boundaries. Final classification describes whether a given tree segment is classified as spruce (green color) or birch (yellow color). Segment class is based on probability densities of the block classifications (Fig. 7).

TABLE I
RESULTS OF DETECTION AND CLASSIFICATION.

	Detection		Classification	
Image	Spruce	Birch	Spruce	Birch
1	4/5	2/2	4/4	1/2
2	4/6	2/2	2/4	2/2
3	3/5	5/5	2/3	3/5
4	5/6	2/4	4/5	2/2
5	4/5	1/2	2/4	1/1
6	3/3	3/3	3/3	2/3
	23/30≈77%	15/18≈83%	17/23≈74%	11/15≈73%

Incorrect classifications were partly due to the imperfect segmentation. Based on block classifications, about 2–3 more trees of both species could have been correctly classified had the block classifications been used to refine the segmentation results. Taking that into account, the classifier still performs worse with testing images than with manually marked segments in training images. Fig. 9 and Fig. 10 demonstrate the potential of refining segmentation with block classifications. In the middle of this location there is a big spruce and multiple birches around it. As is seen in the final classification (Fig. 10), the birches (merged with spruce segments) are classified as spruce even though block classifications (Fig. 9) clearly show the different species.

IV. CONCLUSIONS

This work presented an approach to automatically detect and classify young trees in forest environment. Detection was done with segmentation of laser range images and classification with a two-class Naive Bayes classifier based on image texture features. Overall, results were promising but there are still many issues to be considered, especially on generality, validity and practical implementation of the solution. Methods were kept as simple as possible, aiming to test the suitability of the approach.

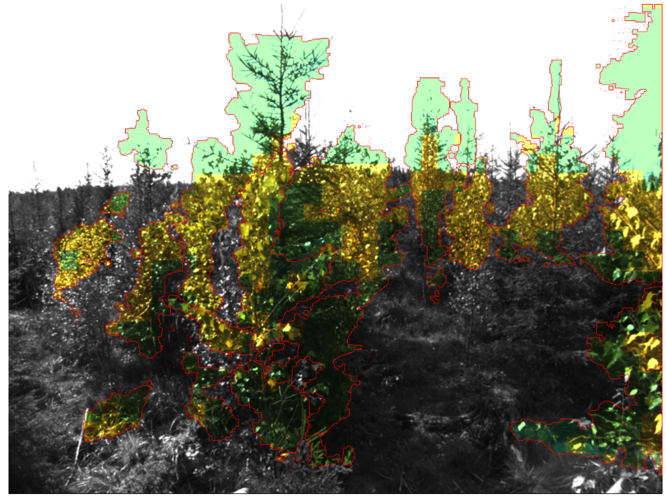

Fig. 9. Block classification for image number 3. Red curves show the segment boundaries. Block classifications describe which parts of the tree segments resemble spruce (green color) and which resemble birch (yellow color). See Fig. 10 for the final classification based on probability densities of the block classifications.

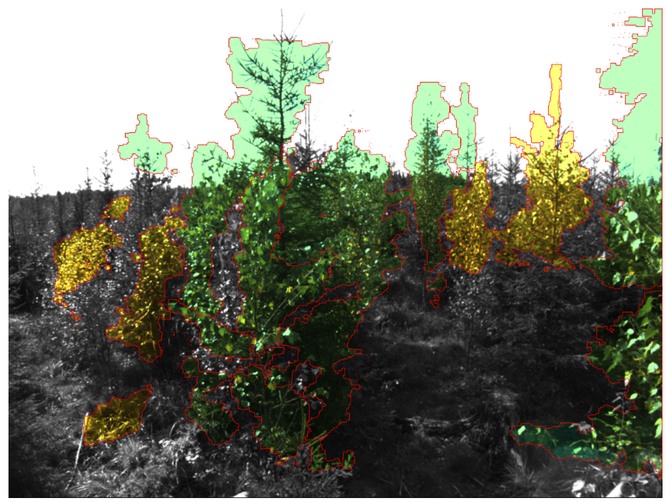

Fig. 10. Final classification for image number 3. Red curves show the segment boundaries. Final classification describes whether a given tree segment is classified as spruce (green color) or birch (yellow color). Segment class is based on probability densities of the block classifications (Fig. 9).

The segmentation method uses constant thresholds to constrain the segment size and the depth difference between points in the segment. It can thus detect only trees of certain size and cannot distinguish between closely packed trees. Segmentation results are reasonable when search points are placed properly and trees are separate enough (see Fig. 4). This is not the case in typical forest environments, however, and many trees in the testing material were split to multiple segments and some distinct trees were combined into single segments. Future work should include finding better growing rules for the region growing algorithm and more sophisticated methods for analyzing each region.

The classification method divides tree segments to image blocks, classifies blocks separately and chooses the tree species based on probability densities of the block classes.

The main issue challenging classification is that features in image blocks are mixed. Some spruce blocks were misclassified due to bright spots caused by bright spruce trunks and birch leaves, whereas birch blocks often included objects well visible through them. Problems with segmentation decrease the classification rates notably. There is a clear need to iterate between segmentation and classification. Comparing images with block classifications (Fig. 7 and Fig. 9) and final classifications (Fig. 8 and Fig. 10), the segmentation result could be improved by noting that block classes form subsegments. On the other hand, measurements taken with the laser scanner could also be used for classification.

The system should be tested with much more data to really assess the performance and reliability. The material consisted only of six training images and six testing images collected in a small forest environment. More images from different locations and weather conditions are needed. Distance from the trees should be varied since it can notably affect texture features. It would be possible to switch between multiple trained classifiers based on distance information. In practice some methods to normalize lighting conditions might also be needed. Histogram equalization was tested but had no notable effect on the result.

Currently only the green channel of the camera images was used giving a better result than standard color-gray transformations. Future work should look for a better way to utilize all the color information available in camera images. Hyyti et al. [1], for instance, used EGRBI color transformation to better separate spruce from the surrounding vegetation. Color information could also be used to separate the sky visible through the trees. The camera should have a higher resolution to compute features more reliably. Only 40% of tree segments detected by the laser scanner projected inside the camera images. Omnidirectional cameras would increase the field of view, however, at the cost of accuracy.

The practicality of the approach is limited because the system cannot be moved while acquiring the 3D measurements. The implementation of algorithms does not work in real time either. The basic laser segmentation and the classification are computationally light operations. Solving problems with split and merged segments, removing ground segments and computing many features takes more time. It might be a better approach to concentrate only on one feature type or perhaps learn some simple hierarchical features. Hyyti et al. [1] achieved a real-time implementation by using this approach and a graphical processing unit.

Another issue is that the approach only forms an initial overview of the area. This reduces the importance of fast processing but also means that other machine perception methods are needed during the actual cleaning operation. Representation of each tree must be made more accurate when the cleaning device starts to move towards the trees.

ACKNOWLEDGMENT

We would like to thank M.Sc. Matti Öhman for designing the 3D laser scanner system, M.Sc. Jakke Kulovesi and M.Sc. Teemu Kemppainen for useful discussions related to texture features and classification, and UPM-Kymmene for allowing us to collect the testing material in their forest.

REFERENCES

[1] H. Hyyti, J. Kalmari, and A. Visala, “Real-time detection of young spruce using color and texture features on an autonomous forest machine,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2013, pp. 2984–2991.

[2] N. Haering and N. Da Vitoria Lobo, “Features and classification methods to locate deciduous trees in images,” *Comput. Vis. Image Und.*, vol. 75, no. 1–2, pp. 133–149, 1999.

[3] S. Kosonen, “Tree species recognition with machine vision using color and texture analysis,” Master’s thesis, Helsinki Univ. Technol., Dept. Autom. & Syst. Technol., 2007.

[4] W. Ali, F. Georgsson, and T. Hellström, “Visual tree detection for autonomous navigation in forest environment,” in *IEEE Int. Veh. Sym.*, 2008, pp. 560–565.

[5] X. Yu, J. Hyyppä, H. Kaartinen, and M. Maltamo, “Automatic detection of harvested trees and determination of forest growth using airborne laser scanning,” *Remote Sens. Environ.*, vol. 90, no. 4, pp. 451–462, 2004.

[6] Å. Persson, J. Holmgren, U. Söderman, and H. Olsson, “Tree species classification of individual trees in Sweden by combining high resolution laser data with high resolution near-infrared digital images,” *Int. Arch. Photogramm., Remote Sens. and Spatial Inf. Sci.*, vol. 36, no. 8, pp. 204–207, 2004.

[7] T. Jääskelainen, R. Silvennoinen, J. Hiltunen, and J. Parkkinen, “Classification of the reflectance spectra of pine, spruce, and birch,” *Appl. Optics*, vol. 33, no. 12, pp. 2356–2362, 1994.

[8] R. Silvennoinen, T. Jääskelainen, K. Nygren, J. Hiltunen, and J. Parkkinen, “Temporal, spatial, and environmental classification of pine reflectance spectra,” *Environ. Sci. Technol.*, vol. 29, no. 6, pp. 1456–1459, 1995.

[9] V. Jokelainen, “Recognition and localisation of young trees using machine perception,” Master’s thesis, Aalto Univ., Dept. Autom. & Syst. Technol., 2010.

[10] M. Erikson and K. Vestlund, “Finding tree-stems in laser range images of young mixed stands to perform selective cleaning,” in *Proc. ScandLaser Scientific Workshop on Airborne Laser Scanning of Forests*, 2003, pp. 244–250.

[11] X. Jiang and H. Bunke, “Edge detection in range images based on scan line approximation,” *Comput. Vis. Image Und.*, vol. 73, no. 2, pp. 183–199, 1999.

[12] S. Mazouzi and Z. Guessoum, “A new distributed approach for range image segmentation,” *Lect. Notes Comput. Sc.*, vol. 7042, pp. 296–303, 2011.

[13] J. Bouguet. (2013) Camera calibration toolbox for matlab. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[14] J. Heikkilä, “Geometric camera calibration using circular control points,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 10, pp. 1066–1077, 2000.

[15] R. Unnikrishnan and M. Hebert, “Fast extrinsic calibration of a laser rangefinder to a camera,” Carnegie Mellon Univ., Tech. Rep., 2005.

[16] R. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[17] X. Tang, “Texture information in run-length matrices,” *IEEE Trans. Image Processing*, vol. 7, no. 11, pp. 1602–1609, 1998.

[18] B. Chaudhuri, N. Sarkar, and P. Kundu, “Improved fractal geometry based texture segmentation technique,” *IEE Proc-E.*, vol. 140, no. 5, pp. 233–241, 1993.

[19] Y. Chen, M. Nixon, and D. Thomas, “Statistical geometrical features for texture classification,” *Pattern Recogn.*, vol. 28, no. 4, pp. 537–552, 1995.

[20] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[21] R. Porter and N. Canagarajah, “Robust rotation-invariant texture classification: Wavelet, gabor filter and gmrf based schemes,” *IEE Proc. Vis. Image Sign.*, vol. 144, no. 3, pp. 180–188, 1997.